

## **CLAIM AMENDMENTS**

### **Claim Amendment Summary**

#### **Claims pending**

- Before this Amendment: Claims 1, 7-35, 37, and 39-47.
- After this Amendment: Claims 1, 7-35, 37, and 39-47.

**Non-Elected, Canceled, or Withdrawn claims:** None.

**Amended claims:** 1 and 46-47.

**New claims:** None.

---

### **Claims:**

**1. (Currently Amended)** A system comprising:

one or more computer-readable media;

an application program interface embodied on the one or more computer-readable media and which can be utilized to present a presentation;

the application program interface comprising:

a plurality of open methods that can be called by an application, the open methods comprising a parameter for specifying a destination for the presentation, a parameter for specifying a topology to be used for the presentation, and at least a parameter for specifying a data source for the presentation; wherein collectively, the parameters for specifying a data source enable data sources to be specified in different manners, wherein

said one parameter for specifying a data source specifies a partial topology object that defines sources, sinks, and transforms to be used in the presentation, wherein the partial topology is converted into a full topology if the application did not provide the parameter for specifying the topology to be used for the presentation as part of the call to the open methods;

a plurality of methods that enable the media engine to configure a media session for a presentation, wherein a media session provides a simple an API for building, configuring, and manipulating a pipeline of components for media flow control between an origin and one or more destinations;

a method for ascertaining the state of [[a]] the media engine that causes the presentation to be presented;

a method for ascertaining capabilities of the media engine;

a method to start processing media samples that are the subject of the presentation;

a method to stop processing media samples; and

a method to pause media sample processing.

## **2. (Cancelled)**

## **3. (Cancelled)**

**4. (Cancelled)**

**5. (Cancelled)**

**6. (Cancelled)**

**7. (Original)** The system of claim 1, wherein one media engine state comprises an initial state that is assumed when the media engine is created.

**8. (Original)** The system of claim 1, wherein one media engine state comprises a connecting state that is assumed when the media engine is trying to open a networked media and is connecting to a server.

**9. (Original)** The system of claim 1, wherein one media engine state comprises an opened state that is assumed when the media engine has successfully opened a media.

**10. (Original)** The system of claim 1, wherein one media engine state comprises a running state that is assumed when the media engine has successfully started a presentation.

**11. (Original)** The system of claim 1, wherein one media engine state comprises a paused state that is assumed when the media engine has successfully paused a presentation.

**12. (Original)** The system of claim 1, wherein one media engine state comprises a shutdown state that is assumed when the media engine is shut down.

**13. (Original)** The system of claim 1, wherein one media engine state comprises a transitioning state that is assumed when an asynchronous call is made on the media engine.

**14. (Original)** The system of claim 1, wherein one media engine state comprises a suspended state that is assumed when the opening or running of the media engine is waiting on a user action.

**15. (Original)** The system of claim 1, wherein the method for ascertaining the capabilities of the media engine returns a bitwise OR of the current capabilities of the media engine.

**16. (Original)** The system of claim 1, wherein the method for ascertaining the capabilities of the media engine returns a bitwise OR of the current capabilities of the media engine, and wherein bits that describe the current capabilities of the media engine comprise a start bit, a skipforward bit, a skip backward bit, a skip node bit, a seek bit, and a pause bit.

**17. (Original)** The system of claim 1, wherein the method to start processing samples comprises a first parameter for specifying a time format to use with other parameters and at least a second parameter for specifying an offset from which to start processing.

**18. (Original)** The system of claim 1 further comprising a method that returns a current destination.

**19. (Original)** The system of claim 1 further comprising a method that returns a presentation clock being used for the presentation.

**20. (Original)** The system of claim 1 further comprising a method that provides access to metadata associated with the presentation.

**21. (Original)** The system of claim 1 further comprising a method that provides access to statistics associated with the presentation.

**22. (Original)** The system of claim 1 further comprising an event generator interface for generating events associated with the presentation.

**23. (Original)** The system of claim 1 further comprising an event generator interface for generating events associated with the presentation, wherein one event comprises an event that is associated with a new presentation.

**24. (Original)** The system of claim 1 further comprising an event generator interface for generating events associated with the presentation, wherein one event comprises an event associated with completion of an open call.

**25. (Original)** The system of claim 1 further comprising an event generator interface for generating events associated with the presentation, wherein one event comprises an event associated with completion of an operation begun by calling said method to start processing media sample.

**26. (Original)** The system of claim 1 further comprising an event generator interface for generating events associated with the presentation,

wherein one event comprises an event associated with completion of an operation begun by calling said method to stop processing media samples.

**27. (Original)** The system of claim 1 further comprising an event generator interface for generating events associated with the presentation, wherein one event comprises an event associated with completion of an operation begun by calling said method to pause processing media samples.

**28. (Original)** The system of claim 1 further comprising an event generator interface for generating events associated with the presentation, wherein one event comprises an event that indicates that a last media sample from an active media source has been rendered.

**29. (Original)** The system of claim 1 further comprising an event generator interface for generating events associated with the presentation, wherein one event comprises an event associated with completion of an operation begun by calling a close method on the media engine.

**30. (Original)** The system of claim 1 further comprising an event generator interface for generating events associated with the presentation, wherein one event comprises an event associated with a presentation switch.

**31. (Original)** The system of claim 1 further comprising an event generator interface for generating events associated with the presentation, wherein one event comprises an event associated with a destination change.

**32. (Original)** The system of claim 1 further comprising an event generator interface for generating events associated with the presentation, wherein one event comprises an event associated with a media engine state change.

**33. (Original)** The system of claim 1 further comprising an event generator interface for generating events associated with the presentation, wherein one event comprises an event that indicates that a set of operations allowed by the media engine has changed.

**34. (Original)** The system of claim 1 further comprising a stream selector interface that provides methods for setting stream selection modes.

**35. (Original)** The system of claim 1 further comprising a stream selector interface that provides methods for setting stream selection modes, wherein one mode comprises an automatic mode in which the media engine is



responsible for selecting which streams are used; another mode comprises a manual mode in which an application has control over which streams are selected.

**36. (Cancelled)**

**37. (Original)** The system of claim 1 further comprising a media session interface that provides methods that enable the media engine to configure a media session for a presentation, wherein one method comprises a method for initializing a full topology on the media session.

**38. (Cancelled)**

**39. (Original)** The system of claim 1 further comprising a media session interface that provides methods that enable the media engine to configure a media session for a presentation, wherein one method comprises a method to start processing media samples for the presentation.

**40. (Original)** The system of claim 1 further comprising a media session interface that provides methods that enable the media engine to configure a media session for a presentation, wherein one method comprises a method to start media sample processing without rendering the media samples.

**41. (Original)** The system of claim 1 further comprising a media session interface that provides methods that enable the media engine to configure a media session for a presentation, wherein one method comprises a method to pause media sample processing in the media session.

**42. (Original)** The system of claim 1 further comprising a media session interface that provides methods that enable the media engine to configure a media session for a presentation, wherein one method comprises a method to stop media sample processing in the media session.

**43. (Original)** The system of claim 1 further comprising a media session interface that provides methods that enable the media engine to configure a media session for a presentation, wherein one method comprises a method to shut the media session down and release resources used by the media session.

**44. (Original)** The system of claim 1 further comprising a media session interface that provides methods that enable the media engine to configure a media session for a presentation, wherein one method comprises a

method to specify a presentation clock to be used in rendering a current media session.

**45. (Original)** The system of claim 1 further comprising a media session interface that provides methods that enable the media engine to configure a media session for a presentation, wherein one method comprises a method to return a presentation clock being used to rendering a current media session.

**46. (Currently Amended)** A system comprising:

- one or more computer-readable media;
- an application program interface embodied on the one or more computer-readable media and which can be utilized to present a presentation;
- the application program interface comprising:
  - a plurality of open methods that can be called by an application, the open methods comprising a parameter for specifying a destination for the presentation, a parameter for specifying a topology to be used for the presentation, and at least a parameter for specify a data source for the presentation; collectively, the parameters for specifying a data source enabling data sources to be specified in different manners, wherein said one parameter for specifying a data source specifies a partial topology object that defines

sources, sinks, and transforms to be used in the presentation, wherein the partial topology is converted into a full topology if the application did not provide the parameter for specifying the topology to be used for the presentation as part of a call to the open methods;

a plurality of methods that enable the media engine to configure a media session for a presentation, wherein a media session provides ~~a simple an~~ API for building, configuring, and manipulating a pipeline of components for media flow control between an origin and one or more destinations;

a method for ascertaining the state of ~~[[a]] the~~ media engine that causes the presentation to be presented;

a method for ascertaining capabilities of the media engine;

a plurality of methods for providing presentation control;

a method that provides access to metadata associated with the presentation;

a method that provides access to statistics associated with the presentation;

an event generator interface for generating events associated with the presentation;

a stream selector interface that provides methods for setting stream selection modes; and

a media session interface that provides methods that enable the media engine to configure a media session for a presentation.

**47. (Currently Amended)** A system comprising:

one or more computer-readable media;

an application program interface embodied on the one or more computer-readable media and which can be utilized to present a presentation;

the application program interface comprising:

an open method that can be called by an application, the open method comprising a parameter for specifying a partial topology object that defines sources, sinks, and

transforms to be used in the presentation; and

a parameter for specifying a full topology to be used for the presentation, wherein the partial topology is converted into a full topology if the application did not provide the parameter for specifying the full topology to be used for the presentation as part of a call to the open method.